

Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks

Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, Shih-Fu Chang

Abstract—In this paper, we study the challenging problem of categorizing videos according to high-level semantics such as the existence of a particular human action or a complex event. Although extensive efforts have been devoted in recent years, most existing works combined multiple video features using simple fusion strategies and neglected the utilization of inter-class semantic relationships. This paper proposes a novel unified framework that jointly exploits the feature relationships and the class relationships for improved categorization performance. Specifically, these two types of relationships are estimated and utilized by rigorously imposing regularizations in the learning process of a deep neural network (DNN). Such a regularized DNN (rDNN) can be efficiently realized using a GPU-based implementation with an affordable training cost. Through arming the DNN with better capability of harnessing both the feature and the class relationships, the proposed rDNN is more suitable for modeling video semantics. With extensive experimental evaluations, we show that rDNN produces superior performance over several state-of-the-art approaches. On the well-known Hollywood2 and Columbia Consumer Video benchmarks, we obtain very competitive results: 66.9% and 73.5% respectively in terms of mean average precision. In addition, to substantially evaluate our rDNN and stimulate future research on large scale video categorization, we collect and release a new benchmark dataset, called FCVID, which contains 91,223 Internet videos and 239 manually annotated categories.

Index Terms—Video Categorization, Deep Neural Networks, Regularization, Feature Fusion, Class Relationships, Benchmark Dataset.

1 INTRODUCTION

VIDEOS carry very rich and complex semantics, and are intrinsically multimodal. Techniques for recognizing high-level semantics in diverse unconstrained videos can be deployed in many applications, such as Internet video search. However, it is well-known that semantic recognition or categorization of videos is an extremely challenging task due to the semantic gap between computable low-level video features and the complex high-level semantics. While significant progress has been achieved in recent years, most state-of-the-art solutions rely on a large set of features with simple fusion strategies to model the high-level semantics. For instance, two popular ways of combining multiple video features are early fusion and late fusion. Early fusion concatenates all the feature vectors into a long representation for classifier training and testing, while late fusion trains a classifier using each feature separately and combines the outputs of all the classifiers. Both methods do not have the capability of explicitly modeling the

correlations among the features, which can be exploited to achieve a better representation. In addition, the existing categorization methods often neglected the relationships of different semantic classes, which can be exploited to boost the categorization performance. Although there exist several works investigating multi-feature fusion or exploiting the inter-class relationships, as will be discussed in the next section, they mostly address the two problems separately.

Motivated by the limitations of the existing techniques and the increasing popularity of using Deep Neural Networks (DNN) for visual categorization, in this paper we propose a novel unified framework that jointly learns the feature relationships and the class relationships using a DNN. Video categorization can also be carried out within the same network utilizing the learned relationships.

Figure 1 gives an overview of the proposed approach. We first extract several popular video features, including the popular frame-based features computed by the convolutional neural networks (CNN) [1], trajectory-based motion descriptors and audio descriptors. The features are then used as the inputs of a DNN, where the first two layers are input and feature transformation layers, respectively. The third layer is called fusion layer, where we impose structural regularization on the network weights to identify and utilize the feature relationships. Specifically, the regularization terms are selected based on two natural prop-

- Y.-G. Jiang, Z. Wu and X. Xue are with School of Computer Science, Fudan University, Shanghai, China. E-mail: {yuj, zwxu, xyxue}@fudan.edu.cn.
- J. Wang is with Institute of Data Science and Technology, Alibaba Group, Seattle, USA. E-mail: wangjun@gmail.com.
- S.-F. Chang is with Columbia University, New York, USA. E-mail: sfchang@ee.columbia.edu.

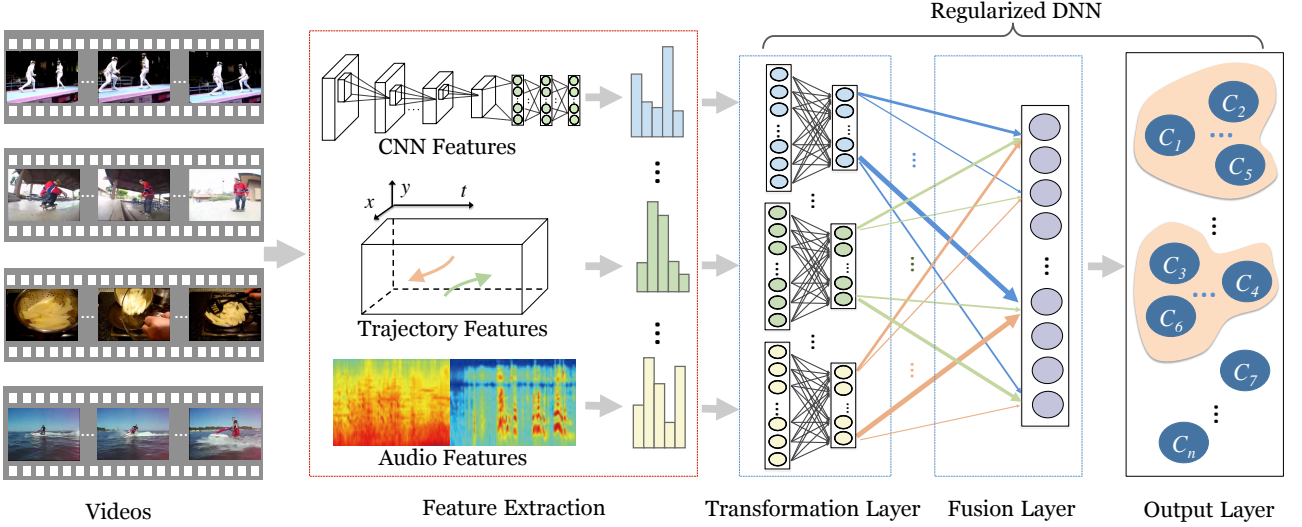


Fig. 1. Illustration of the proposed rDNN framework for video categorization. Various visual and audio features are first extracted and then input into the rDNN. The features are first transformed separately using one layer of neurons. On the fusion layer, regularization on the network parameters is imposed to ensure that different features can share correlated dimensions while preserving some unique characteristics. As shown in the figure, some dimensions of different features are correlated (indicated by the thick lines pointing to the same neuron). After that, the parameters between the fusion and the output layers are also regularized to discover groups of categories. Both the learned feature and class relationships are useful for improving the final performance.

erties of the inter-feature relationships: correlation and diversity. The former means that different features may share some common patterns in a middle level representation lying between the original features and the high-level semantics (i.e., the transformed features after the second layer). The latter emphasizes the unique characteristics of different features, which are the complementary information that is likely to be useful for identifying video semantics. Through modeling both properties using a feature correlation matrix, we impose a trace-norm regularization over the network weights to reveal the hidden correlations and diversity of the features.

In order to discover and utilize the inter-class relationships, we impose similar regularizations on the weights of the final output layer. This helps to identify the grouping structures of video classes and the outlier classes. Semantic classes within the same group share commonalities that can be utilized as knowledge sharing for improved categorization performance, while the outlier classes should be excluded from “negative” knowledge sharing. By executing regularized learning of the two kinds of relationships within the same DNN, we arrive at a unified framework, namely *regularized DNN* (rDNN), which can be easily implemented using a modern GPU.

Although the framework shown in Figure 1 is built on the static CNN feature and a few typical hand-crafted video features, it is feasible to use our approach for fusing any features. We also realize that, in the image categorization domain, the CNN features are dominating state-of-the-art approaches. The

reasons of considering both the CNN feature and the hand-crafted features in this work are two-folds. First, the hand-crafted features have been widely used for video categorization and remain the key components of many systems that generated recent state-of-the-art results on tasks like human action recognition [2] and complex event recognition [3], [4]. By using these features it is easy to make comparisons with the traditional approaches. Second, so far, no existing work on neural networks based video feature extraction has demonstrated significantly better results than the traditional hand-crafted features. Two recent works only showed lower or similar results [5], [6]. Therefore, this paper considers both the deeply learned and the hand-crafted features, and focuses on the tasks of feature fusion and semantic categorization.

The main contribution of this paper is the proposal of the rDNN. To the best of our knowledge, rDNN is the first attempt to exploit both the feature and the class relationships in the DNN pipeline for video categorization. Our formulation is designed to model the complex relationships such as feature/class correlation and diversity. It is easy to implement and can be efficiently executed using a GPU. In addition, realizing the fact that the existing datasets for video categorization are small (e.g., the UCF101 [7]) or lack accurate manual labels (e.g., the DeepSports [5]), we introduce and release a new benchmark dataset, called Fudan-Columbia Video Dataset (FCVID). FCVID contains 91,223 YouTube videos and 239 manually annotated categories. It is one of the largest manually annotated datasets of Internet videos. We evaluate

rDNN using this new dataset, and hope that its public release could stimulate future research around this challenging problem. This work extends upon a conference publication [8] by adding more detailed discussions throughout the paper, introducing the FCVID dataset, conducting new experiments with the CNN feature, and comparing with more alternative methods.

The rest of this paper is organized as follows. Section 2 discusses related works, where we mainly focus on the existing works exploiting feature or class relationships. Section 3 elaborates the proposed rDNN approach. Extensive experimental results and comparisons with several baseline methods and recent state of the arts are discussed in Section 4, where we also provide a brief introduction of the new FCVID dataset. Finally, Section 5 concludes this paper.

2 RELATED WORK

Video categorization has received significant research attention. Most approaches followed a very standard pipeline, where various features are first extracted and then used as inputs of classifiers. Many works have focused on the design of novel features, such as the Spatial-Temporal Interest Points (STIP) [9], trajectory-based descriptors [2], audio clues [10], and the Convolutional Neural Networks (CNN) based features [1], [5], [11], [6].

In contrast to the variety of video features, Support Vector Machines (SVM) have been the dominate classifier option for over a decade. Recently, with the increasing popularity of the deep learning based approaches, neural networks have also been adopted for video classification [5], [11], [6]. Among them, the best deep learning based video categorization result was probably from Simonyan and Zisserman [6], who used a two-stream CNN approach to extract features from static frames and motion optical flow respectively. The features were classified separately and the predictions were then linearly fused. Using this pipeline, they reported similar performance to the improved dense trajectories [2], one of the best hand-crafted feature-based approaches. Besides accuracy, efficiency is another important factor that should be considered in the design of a modern video classification system. Several recent studies investigated this issue by proposing efficient classification methods [12], [13] or parallel computing strategies [14], [15].

In the following we primarily discuss works on multi-feature fusion and/or exploiting class relationships, which are more closely related to this work.

2.1 Exploiting Feature Relationships

In most state-of-the-art video categorization systems, two naive feature fusion strategies were adopted, i.e., the early fusion and the late fusion. Although

both methods cannot exploit the hidden feature relationships like the correlations of different feature dimensions, they are widely used due to simplicity and good generalizability. Fusion weights are needed in both methods to weigh the importance of each individual feature dimension, which can be set as equal values (a.k.a. average fusion) or learned based on cross validation. In several recent works, multiple kernel learning (MKL) [16] was adopted to estimate the fusion weights [17], [18]. MKL was reported to produce better performance in some cases, but the gain was also often observed to be insignificant [19].

Several more advanced feature fusion approaches were recently proposed. In [20], Ye et al. proposed an optimization framework, called robust late fusion, which uses a shared low-rank matrix to remove noises in certain feature modalities. This requires to iteratively compute the singular value decomposition, and therefore is less scalable for large scale applications in high dimensional spaces. In another work by Liu et al. [21], dynamic fusion was adopted to identify the best feature combination strategy for each sample. This approach was shown to be effective but is extremely expensive. In [22], Jiang et al. focused on exploiting the correlations between audio and visual features. They proposed to generate an audio-visual joint codebook by discovering the correlations of the two features for video classification. The approach represents a promising direction as this is one of the very few works performing deep exploitation of feature correlations. The visual features used in this work, however, were computed on the segmented patches of video frames, which is computationally expensive as segmentation is not an easy task. The work was further extended in [23], where the temporal interaction of the audio and visual features was exploited. Jhuo et al. [24] also followed a similar framework, and improved the speed of training the audio-visual codebook by replacing the segmentation-based region features with standard local features like the SIFT [25].

With the growing popularity of the DNN, a few recent studies focused on combining multiple features in neural networks, which are closely related to this work. A deep de-noised auto-encoder was employed in [26] to learn a shared representation based on multimodal inputs. Similarly, a deep Boltzmann machine was utilized in [27] to fuse visual and textual features. Very recently, Kihyuk et al. [28] proposed to learn a good shared representation by minimizing variation of information, so that missing input modality can be better predicted based on the available information. They showed that this method outperforms [27] on several image classification benchmarks. Different from [26], [27] that fused the features in a “free” way without imposing any learning or optimization process, in this paper we propose *regularized fusion* of multiple features, which is intuitively reasonable and

empirically effective. Compared with [28], our objective is to identify dimension-wise feature correlations. Minimizing the variation of information in [28] might be more suitable for images, but for videos, different modalities (e.g., audio and visual) may represent very distinctive information and simply minimizing their variation may not be a good strategy to exploit the complementary information.

2.2 Exploiting Class Relationships

Many researchers have investigated class relationships, commonly termed context, to improve classification performance. In [29], Torralba et al. discussed the importance of context in the task of object detection in images. In [30], [31], the class co-occurrence context was utilized to improve object recognition accuracy. For video classification, Jiang et al. [32] proposed a semantic diffusion algorithm to harness the class relationships. The algorithm has the capability of domain adaptation. In other words, it can adjust pre-defined class relationships based on data distribution of different domain from the training set. Weng et al. [33] proposed a similar domain-adaptive method that not only used the class relationships, but also explored temporal context information of broadcast news videos. Recently, Deng et al. [34] proposed Hierarchy and Exclusion (HEX) graphs, which can capture not only the co-occurrence class relationships, but also mutual exclusion and subsumption. Another two recent works [35], [36] utilized the co-occurrence statistics to help video classification, where the co-occurrence of classes was used more as a semantic feature representation.

Most of these approaches, however, rely on the co-occurrence statistics of the video classes, and thus cannot be used in the cases where the classes share certain commonalities but do not explicitly co-occur in the same video. Our approach can automatically learn and utilize such commonalities using a regularized DNN with a rigorous formulation.

Our formulation is partly inspired by recent research on Multiple Task Learning (MTL) [37], [38]. MTL trains multiple class models simultaneously and boosts the performance of a task (a classifier model) by seeking help from other related tasks. MTL has demonstrated good results in many applications, such as disease prediction [39], [40] and financial stock selection [41]. Sharing certain commonalities among multiple tasks is the key idea of MTL and several algorithms have been proposed with regularizations on the shared patterns across tasks [42], [43], [44]. These works exploited the class relationships in classification or regression problems using the conventional learning approaches, but never injected such regularizations into the DNN.

In fact, neural network is one of the earliest MTL models [45]. See Figure 2(b) for a standard network

structure. In the network, each unit of the output layer refers to a task (class) and neurons of the hidden layers can be viewed as the shared common features. In this paper, we show that, by imposing explicit forms of regularizations, the class relationships can be better exploited for video classification, and thus superior performance over the traditional neural network with implicit task sharing can be attained.

3 REGULARIZED DNN

In this section, we elaborate the details of the proposed regularized DNN for video classification. We start from introducing the notations and settings.

3.1 Notations and Settings

We have a training set with a total of N video samples, which are associated with C semantic classes. Since a video sample may have M types of feature representations (e.g., multiple visual and audio clues), we can use an $(M + 1)$ -tuple to represent each video as:

$$(\mathbf{x}_n^1, \dots, \mathbf{x}_n^m, \dots, \mathbf{x}_n^M, \mathbf{y}_n), n = 1, \dots, N.$$

Here \mathbf{x}_n^m represents the m -th feature of the n -th video sample, and $\mathbf{y}_n = [y_{n1}, \dots, y_{nc}, \dots, y_{nC}]^\top \in \mathbb{B}^C$ is the associated semantic label for the n -th sample. If the n -th sample belongs to the c -th semantic class, the c -th element is set as $y_{nc} = 1$, otherwise $y_{nc} = 0$. The objective for video classification under the above setting is to train prediction models that can categorize new test videos into the C semantic classes.

Simply, one can independently train one classifier for each semantic class, where different features can be combined using either the early or the late fusion scheme. However, such an independent training strategy does not exploit the feature or the class relationships and it often requires a large amount of training samples for each video class. To address these limitations, we propose a DNN framework with structure regularization to perform video classification. In particular, this regularized DNN carries out *feature fusion* with an additional layer, namely fusion layer, to exploit the correlation and diversity of multiple features, as illustrated in Figure 1. Furthermore, we impose additional regularization on the prediction layer to enforce *knowledge sharing* across different semantic classes. With such a regularized DNN framework, we are able to explicitly explore both types of relationships in a uniform learning process. To address the details of the proposed regularized DNN, below we first introduce the background of training standard DNNs with a single type of feature.

3.2 DNN Learning with A Single Type of Feature

Inspired by the biological neural systems, DNN uses a large number of interconnected neurons and construct

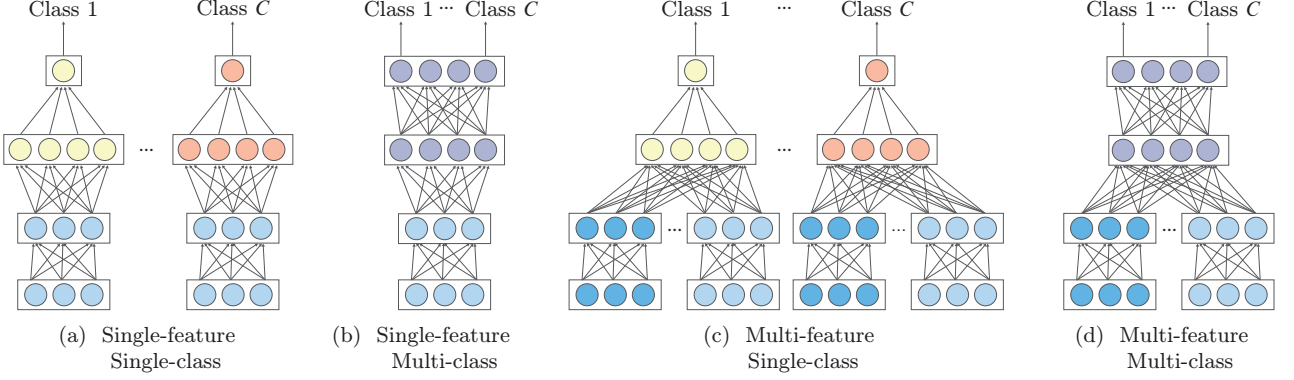


Fig. 2. Popular neural network structures: (a) is the standard one-vs-all training scheme using a single type of feature; (b) is the popular structure used in multi-class learning with a single type of feature; (c) is the one-vs-all training scheme using multiple types of features; and (d) is a recently proposed neural network structure that processes multiple features separately and then performs fusion using a middle layer [27].

complex computational models to mimic the information processing in neural systems. Through cascading the neurons in multiple layers, DNN exhibits strong non-linear abstraction capacity and is able to learn arbitrary mapping from inputs to outputs as long as being given sufficient training data.

Given a DNN with L layers, we denote \mathbf{a}_{l-1} and \mathbf{a}_l as the input and the output of the l -th layer, $l = 1, \dots, L$, while \mathbf{W}_l and \mathbf{b}_l refer to the weight matrix and the bias vector of the l -th layer, respectively. With only a single type of feature, the transition function from the $(l-1)$ -th layer to the l -th layer can be written as:

$$\mathbf{a}_l = \begin{cases} \sigma(\mathbf{W}_{l-1}\mathbf{a}_{l-1} + \mathbf{b}_{l-1}) & l > 1; \\ \mathbf{x} & l = 1, \end{cases} \quad (1)$$

where the nonlinear sigmoid function $\sigma(\cdot)$ is defined as:

$$\sigma(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}}.$$

For simplicity, we can absorb \mathbf{b}_{l-1} into the weights coefficient \mathbf{W}_{l-1} by adding an additional dimension to the feature vectors with a constant value one. Figure 2 (a) and (b) show two types of four-layered neural networks using a single feature as the input to classify data samples into C semantic classes.

Typically, one can minimize the following cost function to derive the optimal weights for each layer in the network:

$$\min_{\mathbf{W}} \sum_{i=1}^N \ell(f(\mathbf{x}_i), \mathbf{y}_i) + \frac{\lambda_1}{2} \sum_{l=1}^{L-1} \|\mathbf{W}_l\|_F^2. \quad (2)$$

The first part in the above cost function measures the empirical loss on the training data, which summarizes the discrepancy between the outputs of the network $\hat{\mathbf{y}}_i = \mathbf{a}_L = f(\mathbf{x}_i)$ and the ground-truth labels \mathbf{y}_i . The second part is a regularization term preventing overfitting.

3.3 Regularization for Feature Fusion

The DNN using a single type of feature is effective in some cases. However, for data with a variety of representations like videos, the semantics could be carried by different feature representations. For instance, some video semantic classes strongly link to the visual effects and others are more relevant to the audio clues. Simple fusion strategies like the early or the late fusion usually have limited performance improvement since the intrinsic relations among multiple feature representations are often overlooked [46]. In addition, such simple fusion methods often incur extra efforts for training the classifiers. Therefore, it is desired to obtain a compact yet effective fused representation that leverages the complementary clues from various features.

Motivated by the multisensory integration process of primary neurons in biological systems [47], [48], we extend the basic DNN with structure regularization on an additional fusion layer to accommodate the *deep fusion* process of multiple types of features. As demonstrated in Figure 1, the fusion layer absorbs all the outputs from the transformation layer to generate an integrated representation as the input for the classification layer. Accordingly, the transition equation for this fusion layer can be written as the following:

$$\mathbf{a}_F = \sigma \left(\sum_{m=1}^M \mathbf{W}_E^m \mathbf{a}_E^m + \mathbf{b}_E \right). \quad (3)$$

We denote E as the index of the last layer of feature transformation and F as the index of the fusion layer (i.e., $F = E + 1$). Hence, \mathbf{a}_E^m represents the extracted mid-level representation for the m -th feature. From the above transition equation, the mid-level representation is first linearly transformed by the weight matrix \mathbf{W}_E^m and then non-linearly mapped to generate the fused representation \mathbf{a}_F using a sigmoid function.

Since the M feature representations reveal various perspectives of the same video data, it is understandable that all these features can be used to learn the common latent semantic patterns. In addition, different types of features can also be complementary to each other since they have distinct clues and characteristics. Hence, the objective for the fusion process should capture the relations among the features, while being able to preserve their uniqueness. Different from most of the straightforward fusion strategies, we specifically formulate an optimization problem with structure regularization on the weights of the fusion layer. Such a regularized DNN encourages the fusion process to explore correlations and diversities among the multiple features simultaneously.

Note that the weights of the fusion layer, $\mathbf{W}_E^1, \dots, \mathbf{W}_E^M$, transform all the available features into a shared representation. Here the weight matrices are first vectorized into P dimensional vectors separately with $P = |\mathbf{a}_E^m| \cdot |\mathbf{a}_F|$ being the product of the \mathbf{a}_E^m 's ($m = 1, \dots, M$) dimension and the \mathbf{a}_F 's dimension. To simplify the formulation, we assume the extracted features \mathbf{a}_E^m are of the same dimension. Then all the coefficient vectors are stacked into a matrix $\mathbf{W}_E \in \mathbb{R}^{P \times M}$. Each column of \mathbf{W}_E corresponds to the weights of a single feature with the element $\mathbf{W}_E(i, j)$ given as

$$\mathbf{W}_E(i, j) = \mathbf{W}_E^i(j), \quad i = 1, \dots, M, \quad j = 1, \dots, P.$$

In order to perform *feature fusion* by exploring correlations and diversities simultaneously, we formulate the following regularized optimization problem to learn the weights of the DNN:

$$\begin{aligned} \min_{\mathbf{W}, \Psi} \quad & \mathcal{L} + \frac{\lambda_1}{2} \left(\sum_{l=1}^E \sum_{m=1}^M \|\mathbf{W}_l^m\|_F^2 + \sum_{l=F}^{L-1} \|\mathbf{W}_l\|_F^2 \right) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}_E \Psi^{-1} \mathbf{W}_E^T) \\ \text{s.t.} \quad & \Psi \succeq 0, \end{aligned} \quad (4)$$

where $\mathcal{L} = \sum_{i=1}^N \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i)$ is the empirical loss term. Different from the standard single feature based neural network (cf. Equation 2), we include one additional regularization term in the above cost function with one more variable $\Psi \in \mathbb{R}^{M \times M}$ to model the inter-feature correlation. Note that Ψ a symmetric and positive semidefinite matrix and the last regularization term with the trace norm can help learn the inter-feature relationship [38], [49]. The entries with large values in Ψ indicate strong feature correlations, while small-valued entries denote the diversity among different features since they are less correlated. The coefficients λ_1 and λ_2 balance the contributions from different regularization terms. Finally, we can introduce a joint minimization procedure over the weight matrix \mathbf{W} and the feature correlation matrix Ψ to train the regularized DNN.

3.4 Regularization for Class Knowledge Sharing

As discussed earlier, one can simply adopt the one-vs-all strategy to independently train C classifiers for categorizing video semantics. As illustrated in Figure 2(a) and 2(c), this one-vs-all training scheme with a total of C four-layered neural networks can be applied for both single-feature and multi-feature settings. To train a total of C neural networks separately, a sufficient amount of positive training samples are desired for each video category. In addition, the independent training process completely neglects the knowledge sharing among different semantic categories. However, video semantics often share some *commonality* due to the strong correlations between different semantic categories, which have been observed in previous studies [32], [50], [51]. Therefore, it is critical to explore such a commonality by simultaneously learning multiple video semantics, which can lead to better learning performance [51]. Generally, the commonality among multiple classes is represented by the parameter sharing among different prediction models [52], [53]. In addition, it is fairly natural for DNN to perform simultaneous multi-class training. For example, as seen in Figure 2(b), by adopting a set of C units in the output layer, a single-feature based DNN can be easily extended to multi-class problems.

Motivated by the regularization methods adopted for MTL [52], [53], here we present a regularized DNN that aims at training multiple classifiers simultaneously with deeper exploitation of the class relationships. To enforce class knowledge sharing, we employ the following optimization problem as our learning objective:

$$\begin{aligned} \min_{\mathbf{W}, \Omega} \quad & \sum_{i=1}^N \ell(f(\mathbf{x}_i), \mathbf{y}_i) + \frac{\lambda_1}{2} \sum_{l=1}^{L-1} \|\mathbf{W}_l\|_F^2 \\ & + \lambda_2 \text{tr}(\mathbf{W}_{L-1} \Omega^{-1} \mathbf{W}_{L-1}^T). \\ \text{s.t.} \quad & \Omega \succeq 0. \end{aligned} \quad (5)$$

Although some previous MTL works explore similar regularization in the learning objective, they often assume that the class relationships are explicitly given and are ready for use as prior knowledge [53], [38]. In our formulation, we tend to learn the prediction model as well as the class relationships. In particular, we adopt a convex formulation by imposing a trace norm regularization term over the coefficients of the output layer \mathbf{W}_{L-1} with the class relationships augmented as a matrix variable $\Omega \in \mathbb{R}^{C \times C}$. The constraint $\Omega \succeq 0$ indicates that the class relationship matrix is positive semidefinite since it can be viewed as the similarity measure of the semantic classes. The coefficients λ_1 and λ_2 are regularization parameters that balance the contributions from different regularization terms.

3.5 Final Objective of rDNN

Considering both objectives of feature fusion and class knowledge sharing, we now present a unified DNN formulation that is able to explore both the feature and the class relationships. In this joint framework, one additional layer is employed to fuse multiple features, where the objective is to bridge the gap between low-level features and the high-level video semantics. Then another layer of neurons is stacked over the fusion layer to generate the predictions, where we impose the trace norm regularization over the prediction models to encourage knowledge sharing across different semantic categories. To build such a rDNN, we incorporate both the feature regularization in Equation 4 and the class regularization in Equation 5 to form the following objective:

$$\begin{aligned} \min_{\mathbf{W}, \Psi, \Omega} \quad & \mathcal{L} + \frac{\lambda_1}{2} \left(\sum_{l=1}^E \sum_{m=1}^M \|\mathbf{W}_l^m\|_F^2 + \sum_{l=F}^{L-1} \|\mathbf{W}_l\|_F^2 \right) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}_E \Psi^{-1} \mathbf{W}_E^T) \\ & + \frac{\lambda_3}{2} \text{tr}(\mathbf{W}_{L-1} \Omega^{-1} \mathbf{W}_{L-1}^T), \\ \text{s.t.} \quad & \Psi \succeq 0 \quad \text{tr}(\Psi) = 1, \\ & \Omega \succeq 0 \quad \text{tr}(\Omega) = 1, \end{aligned} \quad (6)$$

where λ_1, λ_2 , and λ_3 are regularization parameters. In the above formulation, two trace-norm regularization terms are tailored for the fusion of multiple features and the exploitation of the class relationships, respectively. In addition, we impose two additional constraints $\text{tr}(\Psi) = 1$ and $\text{tr}(\Omega) = 1$ to restrict the complexity, as suggested in [38]. In the next section, we introduce an alternating optimization strategy to minimize the above cost function with respect to the network weights $\{\mathbf{W}_l\}_{l=1}^L$, the feature relationship matrix Ψ , as well as the class correlation matrix Ω .

3.6 Optimization and Analysis

For the optimization problem in Equation 6, two pairs of variables, i.e., (\mathbf{W}_E, Ψ) and $(\mathbf{W}_{L-1}, \Omega)$, are coupled with each other. Here we adopt an alternating optimization approach to iteratively minimize the cost function with respect to \mathbf{W}_l^m ($l = 1, \dots, L, m = 1, \dots, M$), Ψ and Ω .

We first consider the minimization problem over the network weight matrix \mathbf{W}_l^m with fixed Ψ and Ω . It is easy to see that the original problem is degenerated to the following unconstrained optimization problem:

$$\begin{aligned} \min_{\mathbf{W}_l^m} \quad & \mathcal{L} + \frac{\lambda_1}{2} \left(\sum_{l=1}^E \sum_{m=1}^M \|\mathbf{W}_l^m\|_F^2 + \sum_{l=F}^{L-1} \|\mathbf{W}_l\|_F^2 \right) \\ & + \frac{\lambda_2}{2} \text{tr}(\mathbf{W}_E \Psi^{-1} \mathbf{W}_E^T) + \frac{\lambda_3}{2} \text{tr}(\mathbf{W}_{L-1} \Omega^{-1} \mathbf{W}_{L-1}^T). \end{aligned} \quad (7)$$

Since all the terms in the above cost function are smooth, the gradient can be easily evaluated. Let \mathbf{G}_l^m be the gradient with respect to \mathbf{W}_l^m . We have the following update equation for the weight matrix \mathbf{W}_l^m :

$$\mathbf{W}_l^m = \mathbf{W}_l^m - \eta \mathbf{G}_l^m, \quad (8)$$

where η is the step length of the gradient descent.

We then introduce the solution for minimizing the cost function over Ψ with other variables being fixed. The problem in Equation 6 can be rewritten as:

$$\begin{aligned} \min_{\Psi} \quad & \text{tr}(\mathbf{W}_E \Psi^{-1} \mathbf{W}_E^T), \\ \text{s.t.} \quad & \Psi \succeq 0 \quad \text{tr}(\Psi) = 1. \end{aligned} \quad (9)$$

By adopting the Cauchy-Schwarz inequality, we obtain the analytical solution for the above optimization problem as:

$$\Psi = \frac{(\mathbf{W}_E^T \mathbf{W}_E)^{\frac{1}{2}}}{\text{tr}((\mathbf{W}_E^T \mathbf{W}_E)^{\frac{1}{2}})}. \quad (10)$$

Similarly, we can derive the optimal solution for Ω as:

$$\Omega = \frac{(\mathbf{W}_{L-1}^T \mathbf{W}_{L-1})^{\frac{1}{2}}}{\text{tr}((\mathbf{W}_{L-1}^T \mathbf{W}_{L-1})^{\frac{1}{2}})}. \quad (11)$$

Note that Zhang et al. adopted a similar solution to identify task correlations for a linear kernel based regression and classification problem [38]. However, our method integrates more complex structure regularizations in a neural network architecture, where both the feature and the class relationships are exploited for a completely different application.

In summary, we first estimate the feature and class relationships using the weights in the neural network. The relationship matrices are then utilized in turn to refine the network weights to improve the classification performance. Due to the existence of analytical solutions, we are able to learn the relationship matrices Ψ and Ω in an efficient way. Finally, the training procedure of the proposed rDNN is summarized in Algorithm 1. In each epoch, we need to compute the gradient matrix \mathbf{G}_l^m for updating \mathbf{W}_l^m , and then update the matrices Ω and Ψ . The complexity of calculating the trace norms is the same as that of the ℓ_2 norm. The update of Ω and Ψ requires cubic-complexity operations with respect to the number of features M and the number of video classes C . In practical large scale settings, the values of M and C are often significantly smaller than the number of training samples. Therefore, the training cost of the proposed rDNN is very similar to that of a standard DNN. Our empirical study further confirms the efficiency of our method, as will be discussed later.

Algorithm 1 Training Procedure of rDNN.

Require: \mathbf{x}_n^m : the representation of the m -th feature for the n -th video sample;

\mathbf{y}_n : the semantic label of the n -th video sample;

1: Initialize \mathbf{W}_l^m randomly, $\Psi = \frac{1}{M}\mathbf{I}_M$ and $\Omega = \frac{1}{C}\mathbf{I}_C$, where \mathbf{I}_M and \mathbf{I}_C are identity matrices;

2: **for** $epoch = 1$ to K **do**

3: Back propagate the prediction error from layer L to layer 1 by evaluating the gradient \mathbf{G}_l^m , and update the weight matrix \mathbf{W}_l^m for each layer and each feature as:

$$\mathbf{W}_l^m = \mathbf{W}_l^m - \eta \mathbf{G}_l^m;$$

4: Update the feature relationship matrix Ψ according to Equation 10:

$$\Psi = \frac{(\mathbf{W}_E^T \mathbf{W}_E)^{\frac{1}{2}}}{\text{tr}((\mathbf{W}_E^T \mathbf{W}_E)^{\frac{1}{2}})};$$

5: Update the class relationship matrix Ω according to Equation 11:

$$\Omega = \frac{(\mathbf{W}_{L-1}^T \mathbf{W}_{L-1})^{\frac{1}{2}}}{\text{tr}((\mathbf{W}_{L-1}^T \mathbf{W}_{L-1})^{\frac{1}{2}})}.$$

6: **end for**

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Dataset and Evaluation

We adopt three challenging datasets to evaluate the rDNN, as described in the following.

Hollywood2 [9]. The Hollywood2 dataset is well-known in the area of human action recognition in videos. Collected from 69 Hollywood movies, it contains 1,707 short video clips annotated according to 12 classes: answering phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up. Following [9], the dataset is split into a training set with 823 videos and a test set with 884 videos.

Columbia Consumer Videos (CCV) [54]. The CCV dataset is a popular benchmark on Internet consumer video categorization. It contains 9,317 videos collected from YouTube with annotations of 20 semantic categories, including objects (e.g., “cats”), scenes (e.g., “playground”), and events (e.g., “parade”). Since many categories are events, it requires a joint use of multiple feature clues like visual and audio representations to perform better categorization. The dataset is evenly split into a training set and a test set.

Fudan-Columbia Video Dataset (FCVID). Since both the Hollywood2 and the CCV datasets are small in terms of the number of annotated classes and the number of videos, to substantially evaluate our rDNN, we collect and release a new benchmark, named FCVID¹. This dataset contains 91,223 Internet videos annotated manually according to 239 categories, covering a wide range of topics like social

events (e.g., “tailgate party”), procedural events (e.g., “making cake”), objects (e.g., “panda”), scenes (e.g., “beach”), etc. We divide the dataset evenly with 45,611 videos for training and 45,612 videos for testing. To the best of our knowledge, FCVID is one of the largest datasets for video categorization with accurate manual annotations. Please refer to the appendix for more information of the dataset, including details on the collection and annotation process, statistics, a category hierarchy, as well as other related released resources (e.g., all the computed features used in this work).

For all the three datasets, we adopt average precision (AP) to measure the performance of each category and report mean AP (mAP) as the overall results of all the categories.

4.1.2 Video Features

As aforementioned, we consider both deeply learned features and hand-crafted features in this work.

Static CNN Features. Recently, CNN has exhibited top-notch performance in various visual categorization tasks, particularly in the image domain [55]. We adopt a CNN model pre-trained on the ImageNet 2012 Challenge data, which consists of 1.2 million images and 1,000 concept categories. For a given video frame, we extract a 4,096-d feature representation (CNN- fc_7), which is the output of the 7th fully connected layer as suggested in [56]. Finally, the frame-level features are averaged to generate a video-level representation.

Motion Trajectory Features [2]. The dense trajectory features [2] have been popular for several years, which have exhibited strong performance on various video categorization datasets. Densely sampled local frame patches are first tracked over time to generate the dense trajectories. For each trajectory, four descriptors are computed based on local motion pattern and the appearance around the trajectory, including a 30-d trajectory shape descriptor, a 96-d histogram of oriented gradients (HOG) descriptor, a 108-d histogram of optical flow (HOF) descriptor, and a 108-d motion boundary histogram (MBH) descriptor. Finally, each type of descriptor is quantized into a 4,000-d bag-of-words representation, following the settings of [2].

Audio Features. The audio soundtracks contain very useful clues that can help categorize some video semantics. Two types of video features are considered in this work. The first one is the popular MFCCs (Mel-Frequency Cepstral Coefficients), which are computed over every 32ms time-window with 50% overlap and then quantized into a bag-of-words representation. The second one is called Spectrogram SIFT (sgSIFT) [57], where we transform the 1-d soundtrack of a video into a 2-D image based on the constant-Q spectrogram. Standard SIFT descriptors are extracted from this spectrogram and quantized into a bag-of-words representation.

1. Available at: <http://bigvid.fudan.edu.cn/FCVID/>

All the bag-of-words representations are normalized with RootSift [58], which has been shown to be more suitable for histogram-based features than the conventional L2 normalization. The CNN-based representation is directly used without further normalization.

4.1.3 Alternative Approaches for Comparison

To verify the effectiveness of our rDNN, we compare with the following approaches:

- 1) **DNN**. The same structure with the rDNN, but no regularization is imposed.
- 2) **Early Fusion with Neural Networks (NN-EF)**. All the features are concatenated into a long vector and then used as the input to train a neural network for video categorization.
- 3) **Late Fusion with Neural Networks (NN-LF)**. A neural network is trained using each feature representation independently. The outputs of all the networks are fused to obtain the final categorization results.
- 4) **Early Fusion with SVM (SVM-EF)**. The popular χ^2 kernel SVM is adopted and the features are combined on the kernel level before classification.
- 5) **Late Fusion with SVM (SVM-LF)**. An SVM classifier is trained for each feature and prediction results are then combined.
- 6) **Multiple Kernel Learning (SVM-MKL)**. We perform feature fusion with the ℓ_p -Norm MKL [59] by fixing $p = 2$. MKL is able to learn dynamic fusion weights. For the above EF/LF approaches 1–4, we adopt equal fusion weights.
- 7) **Multimodal Deep Boltzmann Machines (M-DBM)**. M-DBM is a fusion approach proposed in [27], where multiple feature representations are used as the inputs of the Deep Boltzmann Machines.
- 8) **Discriminative Model Fusion (DMF)**. DMF [60] is one of the earliest approaches for exploiting the inter-class relationships. It simply uses the outputs of an initial classifier, e.g., a DNN in our case, as the features to train an SVM model as the second level classifier to generate the final prediction. The second level SVM is expected to be able to learn and use the class relationships.
- 9) **Domain Adaptive Semantic Diffusion (DASD)**. DASD [32] uses a graph diffusion formulation to utilize the inter-class relationships for visual categorization. Similar to DMF, the prediction outputs of a DNN (without the regularizations) are used as the inputs of the DASD in a post-processing refinement step. The approach requires inputs of pre-computed class correlations, which can be estimated based on statistics of label co-occurrences in the training data. Notice that the pre-computed class correlations are not needed by our rDNN, which can automatically learn the relationships.

Among the alternative approaches, 2–7 focus on feature fusion, while the last two focus on the use of the class relationships. All the neural networks based experiments are conducted on a single NVIDIA Tesla K20 5GB GPU with the MATLAB Parallel Computing Toolbox.

4.2 Results and Discussion

We now report and discuss experimental results. In order to understand the contributions of only exploiting the feature and the class relationships, we first test the performance of the rDNN by disabling the regularizations on the output layer and the fusion layer, respectively. This also ensures to make fair comparisons with the alternative approaches. After that, we enable regularizations on both layers and report results of the entire rDNN framework. With this setting, we analyze the effect of the number of training samples, and compare with recent state-of-the-art results. Last, we discuss the computational efficiency of rDNN.

Throughout the experiments, we set the learning rate of the neural networks to 0.7, fix λ_1 to a small value of 3×10^{-5} in order to prevent overfitting, and tune λ_2 and λ_3 in the same range as λ_1 . We adopt the mini batch gradient descent with the batch size being 70 for network training.

4.2.1 Exploiting Feature Relationships

We first report results by only using the fusion layer regularization in our rDNN, namely rDNN-Feature Regularization (rDNN-F). Table 1 shows the results of the individual features, our rDNN-F, and the alternative feature fusion methods. Among the static CNN, motion and audio features, motion is significant better than the other two on Hollywood2 but is slightly worse than the CNN feature on CCV and FCVID. This is due to the fact that many classes in CCV and FCVID (e.g., “baseball” and “desert”) can be recognized by viewing just one or a few discrete frames, but categorizing the Hollywood2 human actions normally requires a sequence of frames with detailed motion clues. In addition, the overall performance on CCV is slightly lower than that on the much larger FCVID. This is because CCV has some highly correlated categories (cf. Figure 5) that are very difficult to be separated. While FCVID also contains similar confusing categories, the percentage of such “difficult” cases is lower as it also has more “easy” categories, and therefore the overall performance is higher.

For the fusion of the three types of features, our rDNN-F achieves the best performance with consistent gains over all the compared methods. Note that, like the “DNN” baseline, the M-DBM approach also utilizes a neural network for feature fusion, but in a *free* manner without explicitly enforcing the use of the

TABLE 1

Performance comparison (mAP) using individual and multiple features with various fusion methods. “rDNN-F” indicates our rDNN focusing only on the exploitation of the feature relationships.

	Hollywood2	CCV	FCVID
Static CNN	40.1%	66.1%	63.8%
Motion	62.4%	60.8%	62.8%
Audio	22.7%	25.9%	26.1%
DNN	64.2%	71.6%	72.1%
NN-EF	63.5%	70.2%	74.7%
NN-LF	60.2%	69.9%	73.8%
SVM-EF	64.1%	71.7%	75.0%
SVM-LF	62.7%	69.1%	72.1%
SVM-MKL [59]	63.8%	71.3%	75.2%
M-DBM [27]	63.9%	71.1%	74.4%
rDNN-F	65.9%	72.9%	75.4%

feature relationships. These results clearly verifies the effectiveness of imposing the proposed fusion regularization method. Notice that, since the Hollywood2 and the CCV datasets have been widely used, an absolute mAP gain of 2% is generally considered as a significant improvement.

Among the alternative approaches, early fusion methods tend to produce better results than late fusion. This is consistent with the observations of several recent works, where early fusion is more popularly adopted [3]. The MKL is even slightly worse than early fusion on Hollywood2 and CCV, indicating that the learned weights do not generalize well to testing data. In addition, for the contribution of the audio feature in the fusion experiments, we observe clearly improvement for the classes with strong audio clues, such as “answering phone”. On the contrary, for classes like “sitting down”, audio features may slightly degrade the result.

4.2.2 Exploiting Class Relationships

Next, we report results of rDNN using only the class relationships, namely rDNN-C. We compare with the DNN baseline with no regularization, DMF and DASD. Results are given in Table 2. rDNN-C outperforms the DNN baseline and the two alternative approaches. Both DMF and DASD use the outputs of the DNN baseline as inputs for context-based refinement. These results corroborate the effectiveness of the class relationship regularization.

Note that the DASD requires pre-computed class relationships as the input, which are estimated based on the label co-occurrences in the training data. This might be the reason that it performs worse than the rDNN-C as the latter automatically learns the commonalities shared among the categories. The learning process can identify not only the categories that co-occur, but also those sharing visual or auditory commonalities but rarely appear together. To verify this, we visualize some found category groups in

TABLE 2

Performance comparison (mAP) with DMF and DASD, which focus on the use of the class relationships.

“DNN” is a baseline without imposing any regularization and “rDNN-C” indicates our rDNN utilizing only the class relationships.

	Hollywood2	CCV	FCVID
DNN	64.2%	71.6%	72.1%
DMF [60]	61.8%	71.1%	72.5%
DASD [32]	64.4%	71.7%	72.8%
rDNN-C	65.1%	72.1%	74.4%

Figure 3. As discussed in Section 3, values in the matrix Ω can indicate the learned relationships among the categories. Hence, we apply the spectral clustering algorithm on Ω to group the categories and visualize several groups having high within-group category similarities. We see that many categories are grouped together because they share certain commonalities, not due to high frequencies of co-occurrence.

4.2.3 Exploiting Both Kinds of Relationships

Finally, we discuss the results of the entire rDNN framework, using both the feature and the class relationships. To better evaluate the effectiveness of rDNN, we plot the performance with different numbers of training samples in Figure 4. Overall, substantial performance gains are attained from the proposed approach. Using regularizations on both kinds of relationships leads to clearly higher performance than imposing the regularization on a single type of relationship. In addition, comparing the results across the three datasets using all the training samples, the improvement from exploiting the class relationships is more significant on FCVID. This is because FCVID contains a much larger number of classes that share commonalities helpful for categorization. Figure 5 further visualizes the confusion matrices of rDNN on the Hollywood and the CCV datasets.

We also observe that the performance gain of rDNN is more significant when the number of training samples is small (except the case of 10 training samples on FCVID, which are too few to distinguish the 239 categories). Under all the settings, the rDNN requires much less training data to achieve comparable results with the non-regularized version, which is a very appealing property desired in practical applications.

4.2.4 Comparison with State of the Arts

We compare rDNN with several state-of-the-art approaches in Table 3. On Hollywood2, our proposed method achieves a very competitive mAP of 66.9%, outperforming most of the compared approaches [61], [62], [2], [63], [64], [8], except a very recent result from Lan et al. [65]. Almost all these approaches are based on the popular dense trajectory features and the SVM classification with the simple early fusion

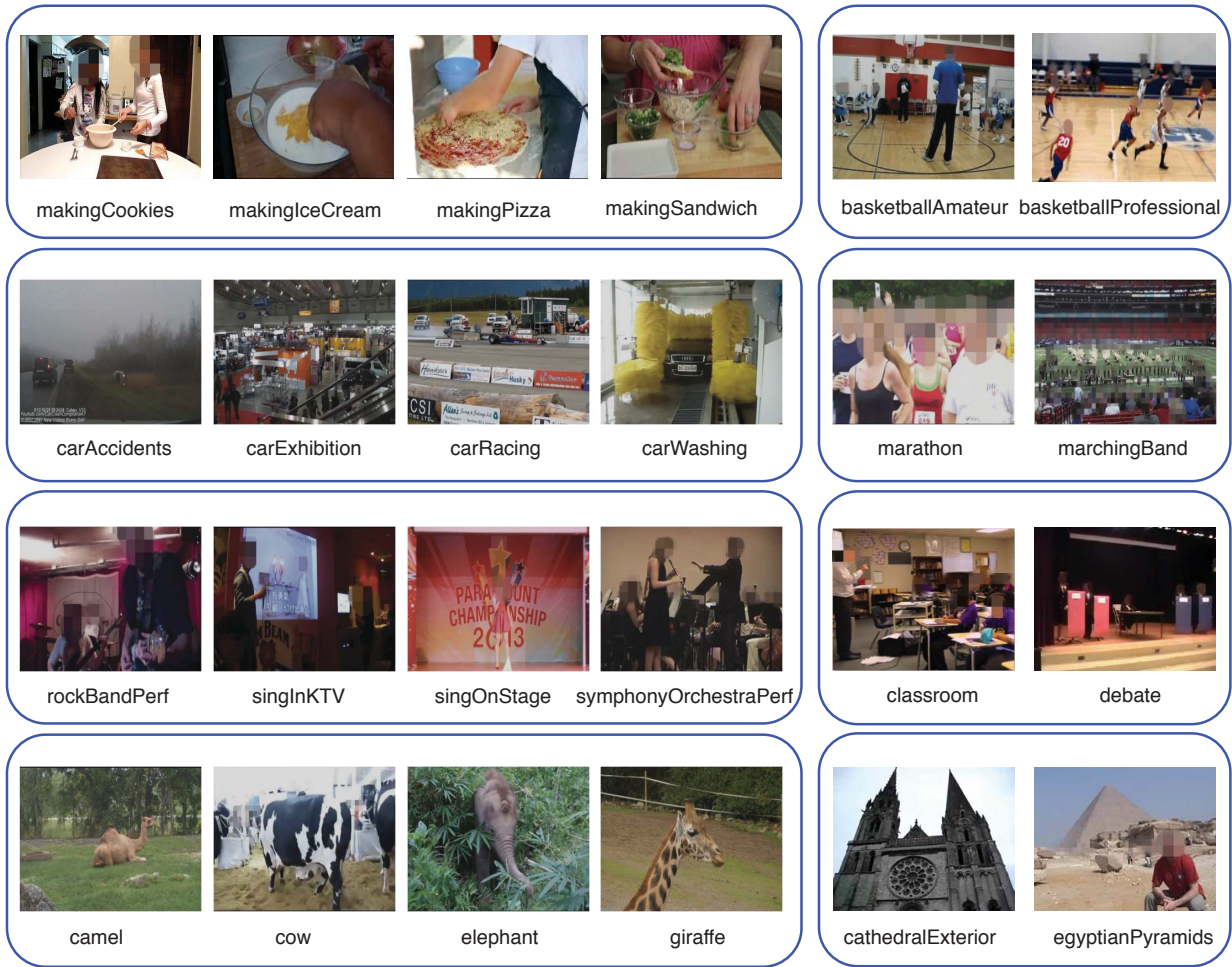


Fig. 3. Example frames of a few automatically found category groups (circled) in the FCVID dataset. Many of the found groups contain categories that share visual/auditory commonalities but do not necessarily co-occur. Discernible faces are blurred due to privacy concern.

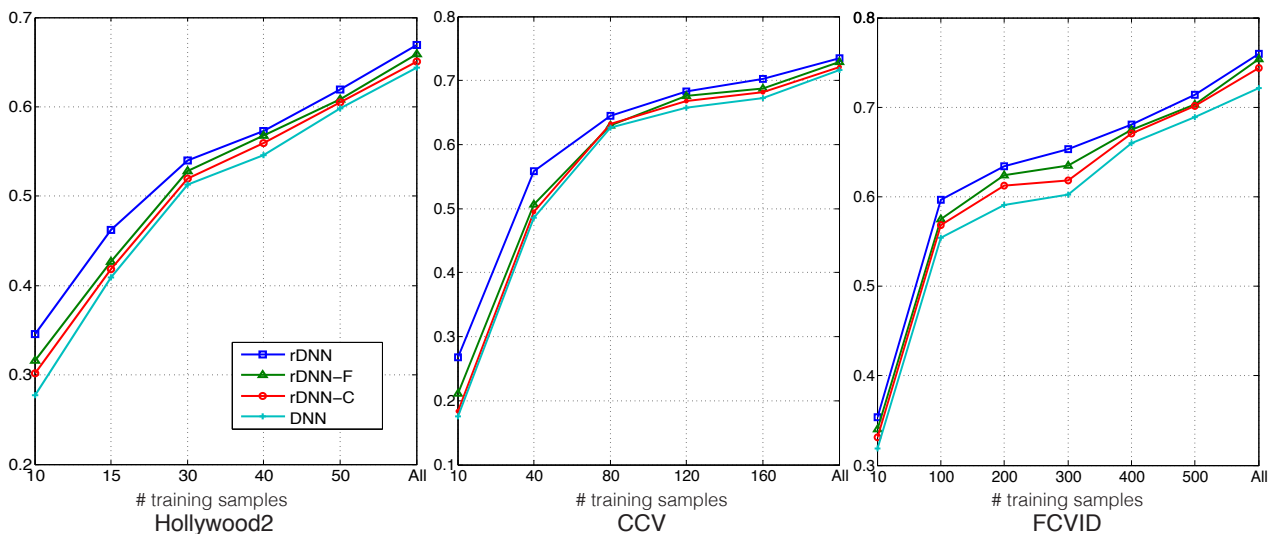


Fig. 4. Performance on the three datasets using different number of training samples. We plot the results of the DNN baseline without regularization, rDNN-F, rDNN-C and the rDNN exploiting both types of relationships. The best mAP on the three datasets (the rDNN approach using all the training samples) are 66.9%, 73.5% and 76.0% respectively. See texts for more discussions.

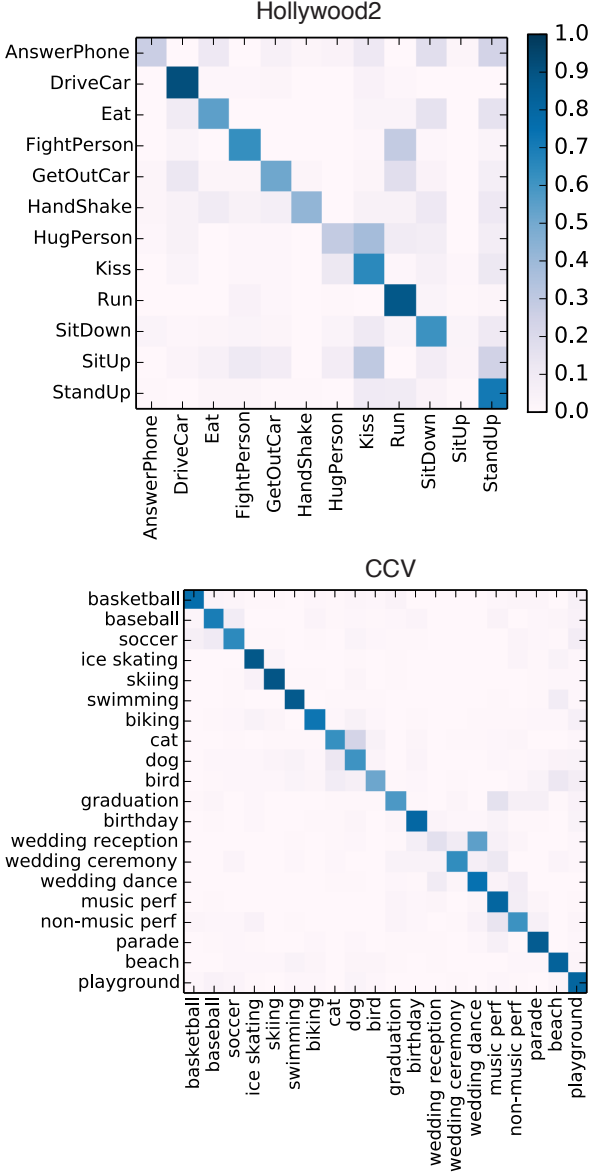


Fig. 5. Confusion matrices of rDNN on Hollywood2 and CCV datasets. Some CCV categories are very confusing, e.g., the three wedding related ones.

method. Note that some of them like Wang et al. [2], Oneata et al. [62] and Lan et al. [65] encoded the features using the Fisher vector [66], which has been shown to be more effective than the classical bag-of-words representation used in our approach. The approach by Lan et al. [65] extends upon the dense trajectories with a feature enhancement method called multi-skip feature stacking. Since our rDNN focuses on feature fusion and classification, we expect that further performance improvements can be achieved by jointly using rDNN with these advanced features.

On the CCV dataset, we obtain to-date the best performance with an mAP of 73.5%. Most recent works on CCV focused on the joint use of multiple audio-visual features. Xu et al. [68] and Ye et

TABLE 3
Comparison with the state of the arts. Our rDNN achieves very competitive results on both the Hollywood2 and the CCV datasets.

Hollywood2	mAP	CCV	mAP
Jain et al. [61]	62.5%	Kim et al. [67]	56.5%
Oneata et al. [62]	63.5%	Xu et al. [68]	60.3%
Wang et al. [2]	64.3%	Ye et al. [20]	64.0%
Zhang et al. [63]	50.9%	Jhuo et al. [24]	64.0%
Ni et al. [64]	61.0%	Ma et al. [69]	63.4%
Wu et al. [8]	65.7%	Liu et al. [21]	68.2%
Lan et al. [65]	68.0%	Wu et al. [8]	70.6%
rDNN	66.9%	rDNN	73.5%

TABLE 4
Training time per epoch of three neural network based approaches, measured on the Hollywood2 dataset.

	Training Time (Seconds)
NN-EF	1.540±0.02
NN-LF	1.552±0.05
rDNN	1.276±0.10

al. [20] extended late fusion with specially designed strategies to remove the noise of individually trained classifiers. Jhuo et al. adopted a joint audio-visual codebook to exploit feature relationships for categorization [24]. rDNN is different from these state-of-the-art approaches and produces significantly better results.

4.2.5 Computational Efficiency

We discuss the computational efficiency of rDNN using the Hollywood2 dataset. The average training time of each epoch for NN-EF, NN-LF and rDNN are given in Table 4, using the same GPU-based implementation. rDNN is more efficient than NN-EF and NN-LF as it contains less parameters to be learned. Specifically, compared with the NN-EF, rDNN processes the features separately in the first two layers and thus avoids the parameters needed for interacting among them. The NN-LF requires the training of separate networks, which is also more expensive. Note that the M-DBM method is not compared because it requires much more time to pre-train the network for weight initialization. For all the methods, normally a few hundreds of epochs are needed to finish the training (several minutes in total). After training, all the neural network based methods are extremely fast in testing.

5 CONCLUSION

We have proposed a novel rDNN approach to exploit both feature and class relationships in video categorization. By imposing trace-norm based regularizations on the specially tailored fusion layer and output layer, our rDNN can learn a fused representation of

multiple feature inputs and utilize the commonalities shared among the semantic classes for improved categorization performance. Extensive experiments of action and event recognition on popular benchmarks have shown that rDNN consistently outperforms several alternative approaches as well as recent state of the arts. Our rDNN is also efficient in both model training and testing, which is very important for large scale applications. In addition, we have introduced a new benchmark dataset, namely FCVID, for large scale video categorization. FCVID is one of the largest datasets in the field with manual annotations, containing 91,223 videos carefully labeled according to 239 categories. We believe that FCVID is helpful for stimulating research not only on video categorization, but also on other related problems.

The current framework supports the use of any pre-computed features. One interesting future work is to exploit the joint learning of feature representations and classification models. For instance, the adopted CNN feature is computed based on off-the-shelf models trained on Image-Net. It would be probably more effective if the feature extraction network could be further tuned simultaneously with the regularized classification network.

APPENDIX

FCVID: FUDAN-COLUMBIA VIDEO DATASET

In this appendix, we introduce the collection and annotation process of the FCVID, and compare it with several existing datasets.

.1 Collection and Annotation

The categories in FCVID cover a wide range of topics like social events (e.g., “tailgate party”), procedural events (e.g., “making cake”), objects (e.g., “panda”), scenes (e.g., “beach”), etc. These categories were defined very carefully. Specifically, we conducted user surveys and used the organization structures on YouTube and Vimeo as references, and browsed numerous videos to identify categories that satisfy the following three criteria: (1) utility — high relevance in supporting practical application needs; (2) coverage — a good coverage of the contents that people record; and (3) feasibility — likely to be automatically recognized in the next several years, and a high frequency of occurrence that is sufficient for training a recognition algorithm.

This definition effort led to a set of over 250 candidate categories. For each category, in addition to the official name used in the public release, we manually defined another alternative name. Videos were then downloaded from YouTube searches using the official and the alternative names as search terms. The purpose of using the alternative names was to expand the candidate video sets. For each search, we downloaded

1,000 videos, and after removing duplicate videos and some extremely long ones (longer than 30 minutes), there were around 1,000–1,500 candidate videos for each category.

All the videos were annotated manually to ensure a high precision of the FCVID labels. In order to minimize subjectivity, nearly 20 annotators were involved in the task, and a master annotator was assigned to monitor the entire process and double-check all the found positive videos. Some of the videos are multi-labeled, and thus filtering the 1,000–1,500 videos for each category with focus on just the single category label is not adequate. As checking the existence of all the 250+ classes for each video is extremely difficult, we use the following strategy to narrow down the “label search space” for each video. We first grouped the categories according to subjective predictions of label co-occurrences, e.g., “wedding reception” & “wedding ceremony”, “waterfall” & “river”, “hiking” & “mountain”, and even “dog” & “birthday”. We then annotated the videos not only based on the target category label, but also according to the identified related labels. This helped produce a fairly complete label set for FCVID, but largely reduced the annotation workload. After removing the rare categories with less than 100 videos after annotation, the final FCVID dataset contains 91,223 videos and 239 categories, where 183 are events and 56 are objects, scenes, etc.

Figure 6 shows the number of videos per category. “Dog” has the largest number of positive videos (1,136), while “making egg tarts” is the most infrequent category containing only 108 samples. The total duration of FCVID is 4,232 hours with an average video duration of 167 seconds. Figure 7 further gives the average video duration of each category.

The categories are organized using a hierarchy containing 11 high-level groups, which is visualized in Figure 8.

.2 Comparison with Related Datasets

We compare FCVID with the following datasets. Most of them have been widely adopted in the existing works on video categorization.

KTH and Weizmann: The KTH [70] and the Weizmann [71] datasets are well-known benchmarks for human action recognition. The former contains 600 videos of 6 human actions performed by 25 people in four scenarios, and the latter consists of 81 videos associated with 9 actions performed by 9 actors.

Hollywood Human Action: The Hollywood dataset [9] contains 8 action classes collected from 32 Hollywood movies with a total of 430 videos. It was further extended to the Hollywood2 [72] dataset, which covers 12 actions from 69 Hollywood movies totaling 1,707 videos. Compared with KTH and Weizmann, where videos were mostly captured under controlled environments with static cameras,

Dataset	# Videos	# Classes	Year of Construction	Background	Manually Labeled?
KTH	600	6	2004	Static	Yes
Weizmann	81	9	2005	Static	Yes
Kodak	1,358	25	2007	Dynamic	Yes
Hollywood	430	8	2008	Dynamic	Yes
Hollywood2	1,787	12	2009	Dynamic	Yes
MCG-WEBV	234,414	15	2009	Dynamic	Yes
Olympic Sports	800	16	2010	Dynamic	Yes
HMDB51	6,766	51	2011	Dynamic	Yes
CCV	9,317	20	2011	Dynamic	Yes
UCF-101	13,320	101	2012	Dynamic	Yes
THUMOS-2014	18,394	101	2014	Dynamic	Yes
MED-2014 (development set)	≈31,000	20	2014	Dynamic	Yes
Sports-1M	1,133,158	487	2014	Dynamic	No
FCVID	91,223	239	2015	Dynamic	Yes

TABLE 5

Several popular benchmark datasets for video categorization, sorted by the year of construction.

the Hollywood datasets are more challenging due to cluttered background and severe camera motion.

Olympic Sports: This dataset was introduced in 2010 [73], containing 800 clips and 16 action classes. The videos were downloaded from the Internet.

HMDB51: The HMDB51 [74] dataset was collected from a variety of sources, such as movies and consumer videos on YouTube. It contains 6,766 videos annotated into 51 classes.

UCF-101 & THUMOS-2014: The UCF-101 [7] dataset is another popular benchmark for human action recognition in videos, which consists of 13,320 video clips (27 hours in total). There are 101 annotated classes that can be divided into five types: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments and Sports. More recently, the THUMOS-2014 Action Recognition Challenge [75] created a benchmark by extending upon the UCF-101 dataset (used as the training set). Additional videos were collected from the Internet, including 2,500 background videos, 1,000 validation videos and 1,574 test videos.

Kodak Consumer Videos: The Kodak consumer videos were recorded by around 100 customers of the Eastman Kodak Company [76]. The dataset consists of 1,358 video clips labeled with 25 concepts (including activities, scenes and single objects) as a part of the Kodak concept ontology [76].

MCG-WEBV: MCG-WEBV is a large set of YouTube videos collected by the Chinese Academy of Sciences [77]. There are 234,414 web videos with annotations on several topic-level events like “a conflict at Gaza”, which are too complicated to be recognized relying merely on content analysis.

Columbia Consumer Videos (CCV): The CCV dataset was constructed in 2011, aiming to stimulate the research on Internet consumer video analysis [54]. It contains 9,317 user generated videos from YouTube, which are annotated into 20 classes, including objects (e.g., “cat” and “dog”), scenes (e.g., “beach” and “playground”), sports events (e.g., “basketball” and “soccer”) and social activities (e.g., “birthday” and “graduation”).

TRECVID MED: Driven by the practical needs of analyzing high-level events in videos, the annual NIST TRECVID activity created a Multimedia Event Detection (MED) task since 2010 [78]. Each year a new or an extended dataset is constructed for worldwide system comparisons. In 2014, the MED dataset contains 20 events, such as “birthday party”, “bike trick”, etc. According to NIST, in the development set, there are around 8,000 videos for training and 23,000 videos as dry-run validation samples (1,200 hours in total). The MED dataset is only available to the participants of the task, and the labels of the official test set (200,000 videos) is not available even to the participants.

Sports-1M: The Sports-1M [5] dataset was released in 2014, consisting of 1 million YouTube videos and 487 classes, such as “bowling”, “cycling”, “rafting”, etc. The dataset is not manually labeled. The annotations were automatically produced by analyzing textual contexts of the videos.

We further summarize and compare FCVID with these datasets in Table 5. FCVID is one of the largest datasets in terms of the numbers of videos and categories. The Sports-1M is larger but focuses only on sports and is not manually labeled.

.3 Released Resources

The dataset can be downloaded after submitting an agreement form. Released resources include the videos, labels, a standard train/test split, a category hierarchy and several pre-computed descriptors (CNN [56], SIFT [79], Improved Dense Trajectories [2], and two audio features). We have also released the textual meta-data (e.g., tags) of the videos to support related research on Internet video analysis. See FCVID website for more details.

ACKNOWLEDGEMENT

We thank Ziqiang Shi, Jiajun Wang, Jian Tu and all the annotators for their help on the collection of the FCVID dataset. We also thank Jian Pu of the Institute of Neuroscience, Chinese Academy of Sciences for helpful discussions in the early stage of this work.

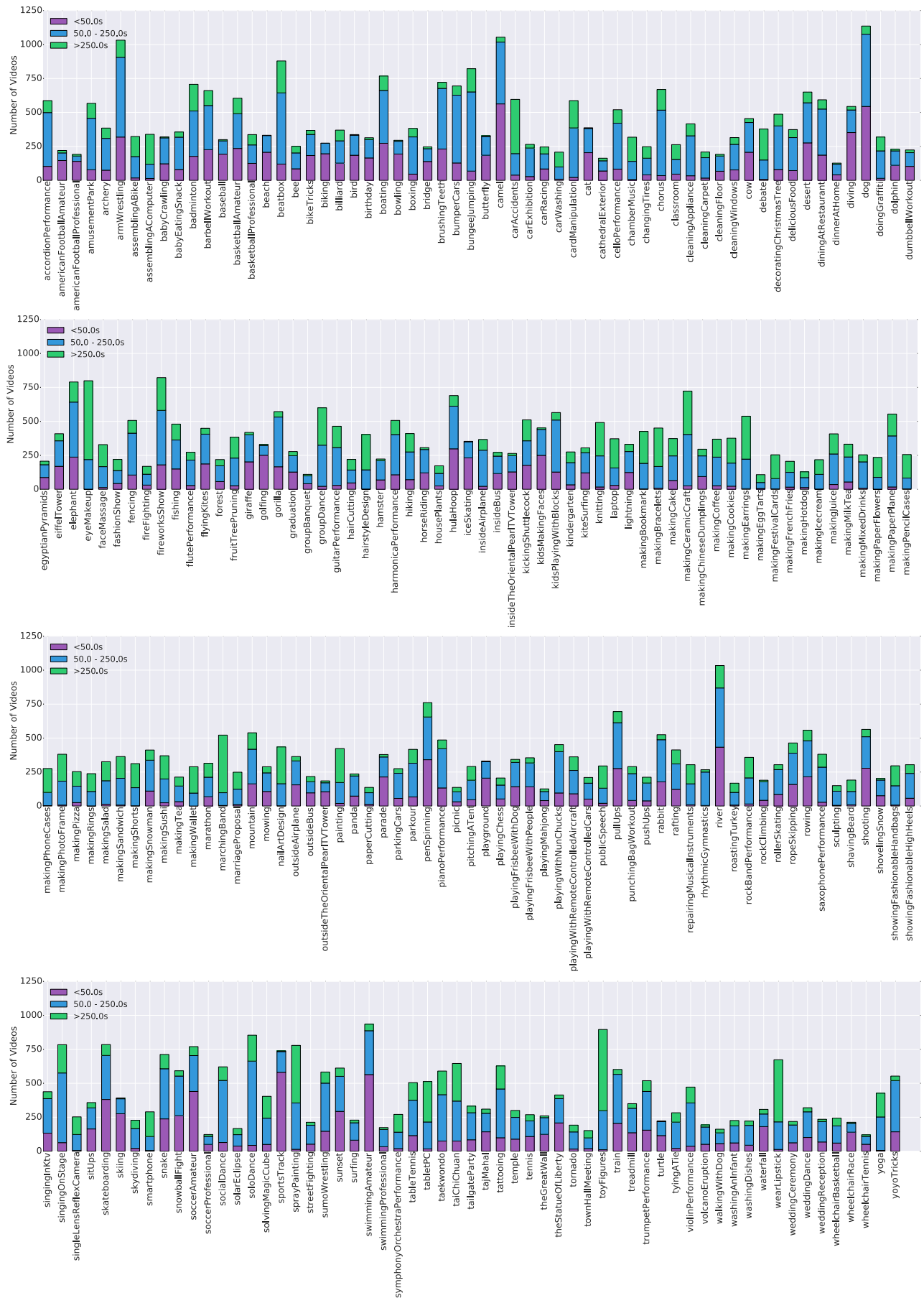
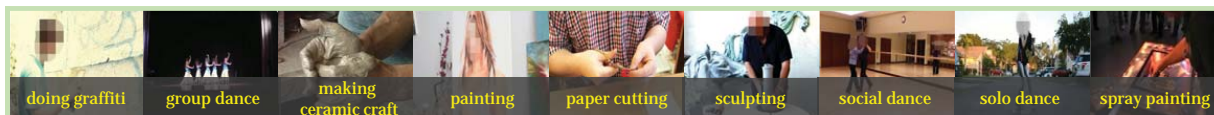


Fig. 6. The number of videos per category in FCVID.



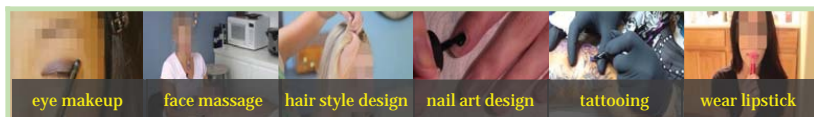
Fig. 7. Average video duration (seconds) per category in FCVID.

Art

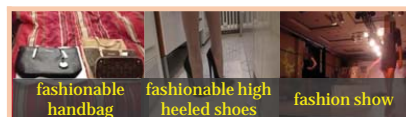


Beauty & Fashion

Beauty



Fashion



Cooking & Health

Drinks



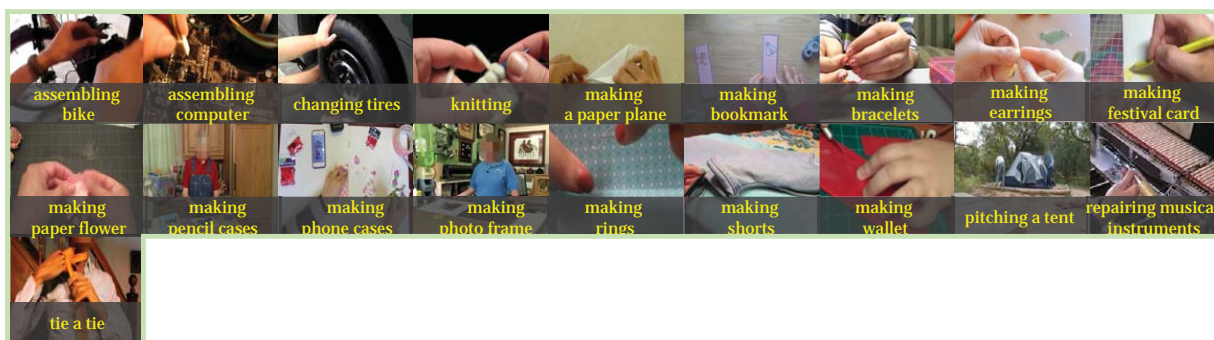
Food



Health

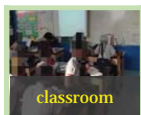


DIY



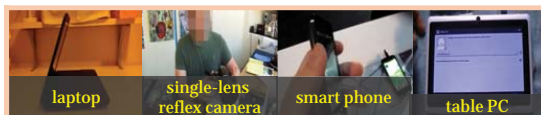
Education & Tech

Education



classroom

High-tech product introductions



laptop

single-lens
reflex camera

smart phone

table PC

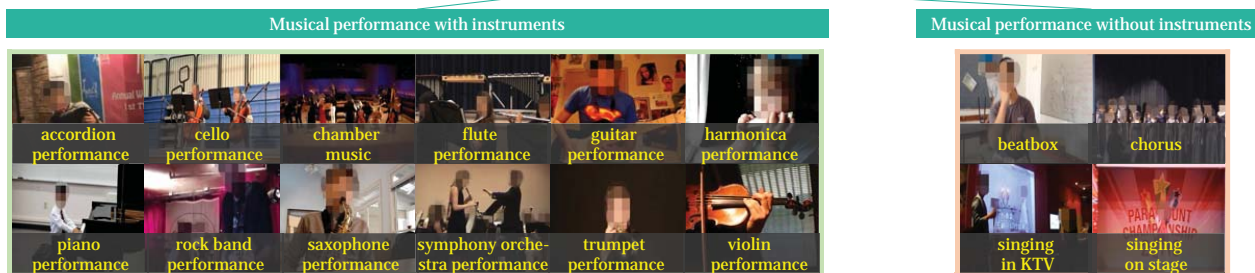
Everyday Life

Activities	Chores	Family Events	Kids	Pests & others	Places	Public Events	Social Events
brushing teeth	car washing	birthday	baby crawling	bird	amusement park	car accidents	dinner at restaurant
hair cutting	cleaning appliance	decorating Christmas tree	baby eating snack	cat	bridge	debate	graduation
parking cars	cleaning carpet	dinner at home	kid playing on playground	delicious food	cathedral	fireworks show	group banquet
shaving beard	cleaning floor		kids making faces	dog	temple	parade	marriage proposal
walking with a dog	cleaning window		kids playing with blocks	hamster		street fighting	picnic
	fruit tree pruning		kindergarten	house plants			tailgate party
	mowing		washing an infant	rabbit			wedding ceremony
	shoveling snow			toy figures			wedding dance
	washing dishes						wedding reception

Leisure & Tricks



Music



Nature

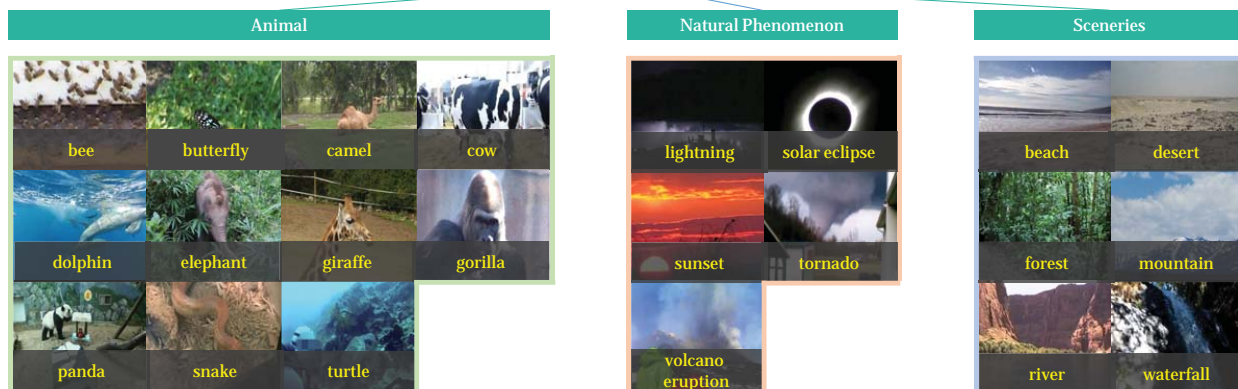




Fig. 8. Visualization of the 239 categories in FCVID, organized in a hierarchy of 11 high-level category groups. The root node is not shown due to space constraint. Discernible faces are blurred.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
- [2] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.
- [3] R. Aly, R. Arandjelovic, K. Chatfield, and et al., "The AXES submissions at TrecVid 2013," in *NIST TRECVID Workshop*, 2013.
- [4] Z.-Z. Lan, L. Jiang, S.-I. Yu, S. Rawat, Y. Cai, C. Gao, S. Xu, H. Shen, X. Li, Y. Wang et al., "CMU-Informedia@TRECVID 2013 multimedia event detection," in *NIST TRECVID Workshop*, 2013.
- [5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.
- [6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.
- [7] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, 2012.
- [8] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue, "Exploring inter-feature and inter-class relationships with deep neural networks for video classification," in *Proceedings of ACM Multimedia*, 2014.
- [9] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.
- [10] C. V. Cotton and D. P. Ellis, "Subband autocorrelation features for video soundtrack classification," in *ICASSP*, 2013.
- [11] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," in *ICML*, 2010.
- [12] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*, 2008.
- [13] Y.-G. Jiang, "Super: Towards real-time event recognition in internet videos," in *ACM ICMR*, 2012.
- [14] Y. Zou, X. Jin, Y. Li, Z. Guo, E. Wang, and B. Xiao, "Mariana: Tencent deep learning platform and its applications," *PVLDB*, 2014.
- [15] O. Yadan, K. Adams, Y. Taigman, and M. Ranzato, "Multi-gpu training of convnets," *CoRR*, 2013.
- [16] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *ICML*, 2004.
- [17] L. Cao, J. Luo, F. Liang, and T. S. Huang, "Heterogeneous feature machines for visual recognition," in *ICCV*, 2009.
- [18] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, and R. Prasad, "Multimodal feature fusion for robust event detection in web videos," in *CVPR*, 2012.
- [19] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009.
- [20] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang, "Robust late fusion with rank minimization," in *CVPR*, 2012.
- [21] D. Liu, K.-T. Lai, G. Ye, M.-S. Chen, and S.-F. Chang, "Sample-specific late fusion for visual category recognition," in *CVPR*, 2013.
- [22] W. Jiang, C. Cotton, S.-F. Chang, D. Ellis, and A. Loui, "Short-term audio-visual atoms for generic video concept classification," in *ACM Multimedia*, 2009.
- [23] W. Jiang and A. C. Loui, "Audio-visual grouplet: temporal audio-visual interactions for general video concept classification," in *ACM Multimedia*, 2011.
- [24] I.-H. Jhuo, G. Ye, S. Gao, D. Liu, Y.-G. Jiang, D. T. Lee, and S.-F. Chang, "Discovering joint audio-visual codewords for video event detection," *Machine Vision and Applications*, 2014.
- [25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.
- [26] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal deep learning," in *ICML*, 2011.
- [27] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *NIPS*, 2012.
- [28] K. Sohn, W. Shang, and H. Lee, "Improved multimodal deep learning with variation of information," in *NIPS*, 2014.
- [29] A. Torralba, "Contextual priming for object detection," *IJCV*, 2003.
- [30] S. Bengio, J. Dean, D. Erhan, E. Ie, Q. Le, A. Rabinovich, J. Shlens, and Y. Singer, "Using web co-occurrence statistics for improving image categorization," *arXiv:1312.5697*, 2013.
- [31] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *ICCV*, 2007.
- [32] Y.-G. Jiang, Q. Dai, J. Wang, C.-W. Ngo, X. Xue, and S.-F. Chang, "Fast semantic diffusion for large-scale context-based image and video annotation," *IEEE TIP*, 2012.
- [33] M.-F. Weng and Y.-Y. Chuang, "Cross-domain multicue fusion for concept-based video indexing," *IEEE TPAMI*, 2012.
- [34] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *ECCV*, 2014.
- [35] S. M. Assari, A. R. Zamir, and M. Shah, "Video classification using semantic concept co-occurrences," in *CVPR*, 2014.
- [36] T. Mensink, E. Gavves, and C. G. M. Snoek, "COSTA: co-occurrence statistics for zero-shot classification," in *CVPR*, 2014.
- [37] L. Jacob, F. Bach, J.-P. Vert et al., "Clustered multi-task learning: A convex formulation," in *NIPS*, 2008.
- [38] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *UAI*, 2010.
- [39] D. Zhang and D. Shen, "Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease," *Neuroimage*, 2012.
- [40] J. Zhou, L. Yuan, J. Liu, and J. Ye, "A multi-task learning formulation for predicting disease progression," in *ACM SIGKDD*, 2011.
- [41] J. Ghosh and Y. Bengio, "Multi-task learning for stock selection," in *NIPS*, 1997.
- [42] J. Chen, J. Zhou, and J. Ye, "Integrating low-rank and group-sparse structures for robust multi-task learning," in *ACM SIGKDD*, 2011.
- [43] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *ICML*, 2011.
- [44] J. Pu, Y.-G. Jiang, J. Wang, and X. Xue, "Multiple task learning using iteratively reweighted least square," in *IJCAI*, 2013.
- [45] R. Caruana, "Multitask learning," *Machine learning*, 1997.
- [46] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, 2010.
- [47] T. Ohshiro, D. Angelaki, and G. DeAngelis, "A normalization model of multisensory integration," *Nature Neuroscience*, 2011.
- [48] B. E. Stein and T. R. Stanford, "Multisensory integration: current issues from the perspective of the single neuron," *Nature Reviews Neuroscience*, 2008.
- [49] H. Fei and J. Huan, "Structured feature selection and task relationship inference for multi-task learning," *Knowledge and information systems*, 2013.
- [50] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *ACM Multimedia*, 2007.
- [51] G. Ye, D. Liu, J. Wang, and S.-F. Chang, "Large-scale video hashing via structure learning," in *ICCV*, 2013.
- [52] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, 2008.
- [53] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l_2 , l_1 -norm minimization," in *UAI*, 2009.
- [54] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *ACM ICMR*, 2011.
- [55] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *CoRR*, 2014.
- [56] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
- [57] B. Zhu, W. Li, Z. Wang, and X. Xue, "A novel audio fingerprinting method robust to time scale modification and pitch shifting," in *ACM Multimedia*, 2010.
- [58] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012.
- [59] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Lp-norm multiple kernel learning," *JMLR*, 2011.
- [60] J. R. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *ICME*, 2003.

- [61] M. Jain, H. Jégou, P. Bouthemy *et al.*, “Better exploiting motion for better action recognition,” in *CVPR*, 2013.
- [62] D. Oneata, J. Verbeek, C. Schmid *et al.*, “Action and event recognition with fisher vectors on a compact feature set,” in *ICCV*, 2013.
- [63] H. Zhang, W. Zhou, C. M. Reardon, and L. E. Parker, “Simplex-based 3d spatio-temporal feature description for action recognition,” in *CVPR*, 2014.
- [64] B. Ni, T. Li, and P. Moulin, “Beta process multiple kernel learning,” in *CVPR*, 2014.
- [65] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, “Beyond gaussian pyramid: Multi-skip feature stacking for action recognition,” *CoRR*, vol. abs/1411.6660, 2014.
- [66] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, “Image classification with the fisher vector: Theory and practice,” *IJCV*, 2013.
- [67] I. Kim, S. Oh, B. Byun, A. A. Perera, and C.-H. Lee, “Explicit performance metric optimization for fusion-based video retrieval,” in *ECCV Workshop*, 2012.
- [68] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. Hauptmann, “Feature weighting via optimal thresholding for video analysis,” in *ICCV*, 2013.
- [69] A. J. Ma and P. C. Yuen, “Reduced analytic dependency modeling: Robust fusion for visual recognition,” *IJCV*, 2014.
- [70] C. Schuldts, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in *ICPR*, 2004.
- [71] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *ICCV*, 2005.
- [72] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” in *CVPR*, 2009.
- [73] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, “Modeling temporal structure of decomposable motion segments for activity classification,” in *ECCV*, 2010.
- [74] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recognition,” in *ICCV*, 2011.
- [75] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, “THUMOS challenge: Action recognition with a large number of classes,” <http://csrcv.ucf.edu/THUMOS14/>, 2014.
- [76] A. Loui, J. Luo, S.-F. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa, “Kodak’s consumer video benchmark data set: concept definition and annotation,” in *ACM MIR Workshop*, 2007.
- [77] J. Cao, Y.-D. Zhang, Y.-C. Song, Z.-N. Chen, X. Zhang, and J.-T. Li, “MCG-WEBV: A benchmark dataset for web video analysis,” *Technical Report, CAS Institute of Computing Technology*, 2009.
- [78] “NIST TRECVID multimedia event detection task 2014,” <http://nist.gov/itl/iad/mig/med14.cfm>, 2014.
- [79] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, 2004.